

日本国特許庁
JAPAN PATENT OFFICE

09.09.03

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: 2002年 9月 9日

出願番号
Application Number: 特願2002-262670
[ST. 10/C]: [JP 2002-262670]

出願人
Applicant(s): ソフトウェア生産技術研究所株式会社

REC'D 23 OCT 2003

WIPO

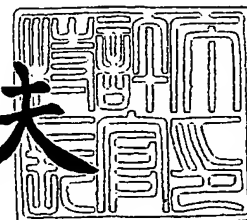
PCT

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH
RULE 17.1(a) OR (b)

2003年10月 9日

特許庁長官
Commissioner,
Japan Patent Office

今井康夫



BEST AVAILABLE COPY

【書類名】 特許願

【整理番号】 2002-095

【提出日】 平成14年 9月 9日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 9/06 530

【発明者】

 【住所又は居所】 東京都大田区南雪谷 4 丁目 2 2 番 1 4 号

 【氏名】 平山 貞宏

【発明者】

 【住所又は居所】 神奈川県鎌倉市十二所 9 6 7 - 6 4

 【氏名】 根来 文生

【特許出願人】

 【識別番号】 599086238

 【氏名又は名称】 ソフトウェア生産技術研究所株式会社

 【代表者】 根来 文生

【代理人】

 【識別番号】 100110559

 【弁理士】

 【氏名又は名称】 友野 英三

【選任した代理人】

 【識別番号】 100110559

 【弁理士】

 【氏名又は名称】 友野 英三

【手数料の表示】

 【予納台帳番号】 164782

 【納付金額】 21,000円

【提出物件の目録】

 【物件名】 明細書 1

 【物件名】 図面 1

【物件名】 要約書 1
【包括委任状番号】 0118513
【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 要件定義方法及びコンピュータソフトウェアの自動開発方法

【特許請求の範囲】

【請求項 1】 (a) 開発対象のコンピュータソフトウェアが最終的に求めようとするアウトプット項目を総て取り出すステップと、

(b) 該取り出されたアウトプット項目の一を計算式及びその成立条件式により規定するステップと、

(c) 前記計算式及びその成立条件式を規定するために現れた新たなデータ項目の総てに対し、当該総ての新たなデータ項目のそれぞれを別の計算式及びその成立条件式により規定するステップと、

(d) 前記ステップ (c) を、当該計算式を構成するのがインプットデータのみとなるまで繰り返すステップと、

(e) 前記ステップ (a) 乃至ステップ (d) を、前記最終的に求めようとするアウトプット項目の総てについて実行し、この実行の結果得られた計算式及びその成立条件式を要件定義とするステップと

を備えることを特徴とする要件定義方法。

【請求項 2】 前記ステップ (b) またはステップ (c) におけるデータ項目は、当該計算式及びその成立条件式に加えて、さらに、インプット／アウトプットの別及びそのデータ項目の存在する記録媒体の明示によっても規定され、

前記ステップ (e) において、前記実行の結果得られた計算式及びその成立条件式に加え、さらに、インプット／アウトプットの別及びそのデータ項目の存在する記録媒体に係る規定をも要件定義とする

ことを特徴とする請求項 1 記載の要件定義方法。

【請求項 3】 (a) 前記請求項 1 もしくは 2 記載の要件定義方法を用いて要件定義を得るステップと、

(b) 前記得られた要件定義中に規定されたデータ項目に基づいて、データの処理順序を自動的に見出すことで或いはデータを自動的に正しい順序で成立させて行くことでプログラムを自動的に開発する手法に適用し、該適用の結果所望のソフトウェアを得るステップと

を備えることを特徴とするコンピュータソフトウェアの自動開発方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、ソフトウェア開発手法に係り、特に、連鎖式データ項目規定法による要件定義方法及びコンピュータソフトウェアの自動開発方法に関する。

【0002】

【従来の技術】

従来のソフトウェア開発手法によれば、基本的に総てのプログラムを、①順序処理、②選択分岐処理、③反復処理の3種類の基本命令文の組み合わせだけで作る。ここで、①順序処理とは、記述されたプログラム文の順にデータ処理を実行する処理をいい、②選択分岐処理とは、条件を判断してデータ処理の実行順序を、記述されたプログラム文の順序から他のプログラム文へ分岐させる処理をいい、③反復処理とは、ある条件が成立している間は記述されたプログラム文を繰り返し実行する処理をいう。

【0003】

これらの命令文は総て処理順序を指示するものであり、総てのプログラムはこれらによってデータの処理順序を人が総て指示しなければならない。データ項目が多かったり実行条件が複雑多岐である錯綜する業務システムでは、プログラミングによって正しい処理順序を指示するためには、人は業務システムをよく理解し総てのデータの処理順序を正しく把握しなければならない。この点、一部のデータ項目だけの処理順序を把握しても、処理順序は全体のデータの関係を決めなければならないので意味がない。

【0004】

このように正しい処理順序でプログラミングすることを最終目的としつつ、そのための準備作業として業務分析、業務デザイン、システムデザイン、プログラムデザイン等が行われ、そこでは例えば下記のようないろいろな工夫がなされている。

・業務システムの中でのデータの流れを示す図としてデータフローダイアグラム

を作成する。

・データ処理の要素機能（例えば販売金額を「単価×数量」として計算する機能）を幾つか含んだデータ処理機能のあるまとまり（例えば個々の販売金額を計算し更にそれらを合計する機能）を正しい処理順序で作る（「モジュール」と呼ぶ）、更にそれらのモジュールを正しい順序で組み合わせることによって全体のプログラムを作る。

- ・データ処理の順序を示すフローチャートを作る。
- ・データ処理の途中で中間データ項目を作って考えを整理する。
- ・データ処理の途中で中間ファイルを作って考える範囲を限定する。

【0005】

従来のソフトウェアの開発方法は、業務システム側から見れば、業務システムの個々の要素機能（例えば販売金額を「単価×数量」として計算する機能）を決め、次にそれらをどんな順序で実行するかを決めてそれらをプログラムでコンピュータに指示するというものである。前述の通りコンピュータは処理順序を自ら探すことはできないので総て人が指示しなければならない。業務の要素機能は、ソフトウェアのユーザである業務部門が決めることであるからユーザズ・ロジック（Users' Logic）と呼び、それらの処理の実行順序をコントロール・ロジック（Control Logic）と呼ぶ。

【0006】

従来のプログラミングとは、端的に言えば、与えられたユーザズ・ロジック（Users' Logic）と正しいコントロール・ロジック（Control Logic）とをコンピュータに指示することである。個々のユーザズ・ロジック（Users' Logic）はユーザが必ず決めなければならないものであるが、これは比較的簡単に決めることができる。プログラミングの作業が膨大になる理由はコントロール・ロジック（Control Logic）を正しく指示することが容易ではないからである。そのための準備や工夫として、先に述べた詳細な業務分析、業務デザイン、データフローダイアグラムの作成、プログラムデザイン、モジュールの作成、フローチャートの作成、中間データ項目の設定、中間ファイルの設定等が行われている。これらは総て正しい順序でプログラムを書くための準備作業である。それらの後

にプログラミングによって正しい処理順序のコントロール・ロジック (Control Logic) を指示する。

【0007】

以上をまとめれば、従来、データ処理順序は人がプログラムによってコンピュータに指示しなければならなかった一方、錯綜したシステムでは、総てのデータの処理順序を人がコンピュータに間違いなく指示することは容易なことではない。このため全体システムの総てのデータの流れを正しく理解把握しなければならなくなり、業務分析、業務デザイン、データフローダイアグラムの作成、プログラムデザイン、等々のプログラミングの前の準備やプログラミングの膨大な作業が必要となってしまう。コストも時間も膨大である。

【0008】

【発明が解決しようとする課題】

上記の問題をさらに詳細に検討する。

【0009】

従来のソフトウェア開発方法は、前述の通りコンピュータがデータの処理順序を自ら見出すことができないので人が総て指示しなければならないが、すこし複雑なシステムでは処理順序は複雑に錯綜しそれらの全体を理解把握することは容易ではない。このことが原因となり下記のような問題が起きている。

- ・ソフトウェア開発に膨大な手間、時間、コストが掛かる。
- ・エンドユーザが業務システムデザインするためにはある程度のシステムエンジニアリングの知識が必要であることから、結局、システムエンジニアに過度に依存し勝ちになる。
- ・実務経験がなく業務の運営・成果責任はないシステムエンジニアがリーダーシップを取り勝ちになる。

【0010】

これらの問題点に加えて、システムエンジニアが主導するので、エンドユーザが本当に望むソフトウェアができるまでにやり直しが多くなる、必要なタイミングを逸することも多い、といった問題点もあった。

【0011】

本発明は、このような従来の技術が有していた諸問題点を解決しようとするものであって、機械的に要件定義を行うことが可能な要件定義方法を提供することを目的とするものである。

【0012】

本発明の更なる目的は、機械的に行われた要件定義とそれに基づいてデータの処理順序を自動的に見出して、或いはデータを自動的に正しい順序で成立させて行って、プログラムを自動的に開発する手法（例えば「L y e e」）を用いることを適合させることによりソフトウェアの開発を自動的に行うというコンピュータソフトウェアの自動開発方法を提供することにある。

【0013】

ここで「L y e e」とは、本願発明に係る発明者の一人である根来文生により発明されたソフトウェア生産方法等に係る発明、技術のことをいい、その詳細は例えば、国際公開番号WO97/16784 A1、WO98/19232 A1、WO99/49387 A1、WO00/79385 A1等によって規定される諸国際公開公報等に開示されている。

【0014】

【課題を解決するための手段】

本発明は上記目的を達成するために、本発明の第1課題解決手段は、（a）開発対象のコンピュータソフトウェアが最終的に求めようとするアウトプット項目を総て取り出すステップと、（b）該取り出されたアウトプット項目の一を計算式及びその成立条件式により規定するステップと、（c）当該計算式及びその成立条件式を規定するために現れた新たなデータ項目の総てに対し、当該総ての新たなデータ項目のそれぞれを別の計算式及びその成立条件式により規定するステップと、（d）ステップ（c）を、当該計算式を構成するのがインプットデータのみとなるまで繰り返すステップと、（e）ステップ（a）乃至ステップ（d）を、該最終的に求めようとするアウトプット項目の総てについて実行し、この実行の結果得られた計算式及びその成立条件式、及び該当する場合にはそれらに加えてさらに、インプット／アウトプットの別及びそのデータ項目の存在する記録媒体の明示による規定を要件定義とするステップとを備えることとしたものであ

る。

【0015】

また、第2の課題解決手段は、(a)第1の課題解決手段に規定される要件定義方法を用いて要件定義を得るステップと、(b)その結果得られた要件定義中に規定されたデータ項目に基づいて、データの処理順序を自動的に見出すことで或いはデータを自動的に正しい順序で成立させて行くことでプログラムを自動的に開発する手法に適用し、該適用の結果所望のソフトウェアを得るステップとを備えることとしたものである。

【0016】

上記第1の課題解決手段による作用は次の通りである。すなわち、本発明の連鎖式データ項目規定による要件定義法によって、目的であるデータ項目が決まりさえすればその規定から出発して簡単に機械的に要件定義を行うことができる。この点従来法では、この方法による要件定義はデータの処理順序を示していないので役立たない。

【0017】

また、上記第2の課題解決手段による作用は、本願発明の第1課題解決手段に係る機械的に要件定義を得、この機械的になされた要件定義に基づいて、データの処理順序を自動的に見出し或いはデータを自動的に正しい順序で成立させて行って、プログラムを自動的に開発する手法（例えば「L y e e」）を用いるので、結果的にでき上がるソフトウェアの開発は自動的とすることが可能となる。

【0018】

本願発明の第1課題解決手段及び第2課題解決手段を用いる結果、データ処理順序を人が考えプログラミングとしてコンピュータに指示するということはしなくてよくなるので、従来法で必要であった詳細な業務分析、業務デザイン、データフローダイアグラムの作成、プログラムデザイン、モジュールの作成、フローチャートの作成、中間データ項目の設定、中間ファイルの設定、さらにはプログラミング自体も人が行う必要がなくなる。従って、従来必要であった膨大なソフトウェア開発作業が大幅に縮減できる。これにより、開発時間とコストが大幅に削減できる。

【0019】

因みに、従来から、データ項目群をそれを構成する個々のデータ項目及びそれと他のデータ項目との関係を示すポインタによって表現したリスト構造というデータ項目群の概念があり、それをベースにしたプログラミング言語がある。これに対して本願発明の第1課題解決手段は、データ項目群を創出するときに個々のデータ項目をデータ項目の規定だけで連鎖的に摘出する方法である。そこには上述のポインタの概念はない。また本願発明の第2課題解決手段は、第1課題解決手段で規定されたデータ項目群に基づき（リスト構造とは関係なくポインタなしで）データ項目の処理順序を自動的に見出して、或いはデータを自動的に正しい順序で成立させて行って、プログラムを自動的に開発する方法を用いることによりソフトウェアを自動的に開発する方法である。

【0020】

【発明の実施の形態】

本願に係る第1の発明（以下、「本願第1発明」ということもある。）について、簡単な例を用いて説明する。例えば販売金額を求めるソフトウェアを開発しようとするとき、先ず求めるアウトプットである販売金額を下記の式のように規定する。次にその計算式に現れた新しいデータ項目（販売単価と販売数量）を夫々に規定する。同様のことを各データ項目がインプットデータと規定されるまで続け総てそうなれば完了する。

- ① 販売金額＝販売単価×販売数量、
- ② 販売単価＝仕入れ単価×（1＋マージン）、
- ③ 販売数量＝インプットデータ、
- ④ 仕入れ単価＝インプットデータ、
- ⑤ マージン＝インプットデータ、

【0021】

因みにそれらの正しい処理順序は、個々のデータ項目の値が得られるためにはその前にそれに必要なデータ項目の値が得られていなければならないから、正しい処理順序について人は③、④、⑤、（この3つについては総てインプットデータであるからそれら間の処理順序は問わない）、②、①の順であることを探り当

ることができる。データ項目が多く、且つ、条件分岐も多い複雑なシステムでも、時間がどれほど掛かってもよければ、人は正しい順序を探り当てることは可能である。しかし残念ながら現在のソフトウェア開発手法では、コンピュータは処理順序を探り当てる機能を持っていない。「従来の技術」項に述べたように、現在のソフトウェア開発の諸問題は、コンピュータのこの処理順序探索不能性から来ているのである。

【0022】

本願第1発明の原理を一般的に述べれば次の様になる。

【0023】

ソフトウェアを開発したいときは必ず、ある求めたいデータ項目が存在する。本願第1発明に係る要件定義法は、一般的に、まずこの目的のデータ項目を決め、次にそのデータ項目の規定をし、ついでその定義式や成立条件の中に新たに出て来た新しいデータ項目を更に規定する。これを連鎖的に次々に行う。最終的に新たなデータ項目の規定がインプットデータとなれば完了する。夫々のデータ項目の規定は、計算式、成立条件、インプット／アウトプットの別、そのデータ項目の存在する記録媒体の夫々の規定によって行う。図1にデータ項目間の関係の例を示す。

【0024】

即ち、図1は、データ項目間の関係を模式的に示すと共に、連鎖式データ項目規定による要件定義法 (Data Chain Requirement Definition (DCRD)法) を概念的に説明するための概念図である。この図はデータ項目間の関係を示すものであって、データ処理の順序を示すデータフローチャートとは全く異なることに注意されたい。

【0025】

同図に示す例にあっては、目的データ項目A (図中の1) は、中間データ項目B (図中の11)、中間データ項目C (図中の12) 及び中間データ項目D (図中の13) により規定されることを示している。さらにその中間データ項目B (図中の11) は、Eファイル111中に格納されるインプットデータ項目E及びFファイル112中に格納されるインプットデータ項目Fにより規定されること

が示される。同様に、中間データ項目 C（図中の 12）は G 画面 121 上に存在するインプットデータ項目 G 及び H 画面 122 上に存在するインプットデータ項目 H により規定され、一方中間データ項目 D（図中の 13）は H 画面 122 上に存在するインプットデータ項目 H 及び I ファイル 131 中に格納されるインプットデータ項目 I により規定されることが、それぞれ示されている。このようなデータ項目間の関係は、通常、種々異なる被開発ソフトウェアの種類、機能、コンピュータ構成等に応じて変化する。E ファイル 111、F ファイル 112、I ファイル 131 及び G 画面 121 並びに H 画面 122 は事実上オプションであり、また、ここには図示されないこれら以外の媒体上にインプットデータ項目が登載されていてもよい。

【0026】

上記のような図 1 の状況下において、ここで具体的に、データ項目間の関係の中で次のような条件分岐があるとする。即ち、

もし $F > G$ なら、 $A = B \times C / D$ を実行する。

そうでなければ、 $C = G / H$ を実行する。（実行すべき式が同じデータ項目 A の定義式、 $A = C + G + H$ である場合でもよい。）

因みに上記を従来法の一般的プログラム言語で書けば、

if $F > G$ then $A = B \times C / D$

else $C = G / H$ （上記カッコの場合、 $A = C + G + H$ ）

【0027】

図 1 のようなデータ項目間の関係と上記の条件分岐がある場合、連鎖式データ項目規定による要件定義法（英語では Data Chain Requirement Definition 法＝「DCRD 法」と名付ける。）は次のようになる。即ち、

$A = B \times C / D$ （定義式の規定）

$F > G$ （成立条件の規定）

アウトプットデータ。（インプット／アウトプットの別の規定）

A 画面にある。（存在する記録媒体の規定）

（ここでの新しいデータ項目は B、C、D、F、G だけ）

ら夫々下記に規定する。以下新しいデータ項目が出て来る毎に規定を行う。)

(上記カッコの場合、

A = C + G + H 。Aの定義式が2つあるがそれでよい。
実行の結果、成立条件によりいずれかの式が成立することになる。)

F > Gではない。

アウトプットデータ。

A画面にある。

ここでの新しいデータ項目は、C、G、H、F、Gだから夫々下記に規定する。)

B = E + F

C = G / H

F > Gではない。(上記カッコの場合、この条件規定は不要。)

D = H - I

E = インプットデータ。Eファイルにある。

F = インプットデータ。Fファイルにある。

G = インプットデータ。G画面にある。

H = インプットデータ。H画面にある。

I = インプットデータ。Iファイルにある。

【0028】

実際の作業においては、例えば、下記の表1に示すような要件定義のためのデータ項目作成表を用いて、現れる各データ項目に対して定義を規定するようにすれば、漏れなく所望の効果を達成することができる。表1はこのようなデータ項目作成表の一である連鎖式データ項目規定による要件定義(Data Chain Requirement Definition)のためのデータ項目作成表であるが、これに関する説明は後述する。

【表 1】

【書項名】 表

【表 1】 連鎖式データ項目規定による要件定義(Data Chain Requirement Definition)のためのデータ項目作成表

データ 項目 番号	データ項目呼称	ID	計算式	成立条件	:	O	記録媒体呼称	記録媒体 ID	備考

【0029】

以上の通り、本願第1発明によれば、ソフトウェアを生産するにおいて、要件定義はまず目的のデータ項目を決め、ついでその定義式や成立条件の中に新たに出て来た新しいデータ項目を更に規定するという方法により、人が処理順序等を考察することなく連鎖的にそこから次々と掘り起こすようにして摘出してしまふ。換言すれば本願第1発明は、ソフトウェア開発において、要件定義をデータ項目の規定だけで行い、かつデータ項目の摘出を連鎖的機械的に簡単に行う方法である。

【0030】

このようにデータ項目によって要件定義を行うためのソフトウェアツールが準備されている。画面上に現れる表1のデータ項目の呼称欄に目的であるアウトプットのデータ項目の呼称、ID（識別符号）、計算式(式中のデータ項目は呼称で書く)、成立条件、インプット(I)/アウトプット(O)の別、そのデータ項目が存在する記録媒体の呼称とIDをインプットすれば、新しい行に新しく計算式や成立条件に現れたデータ項目の呼称と番号が自動的に現れる。新しい行に次々に新しく現れるそれらのデータ項目について一つ一つ規定して機械的に欄を埋めて行けばよい。最終的に新しく現れたデータ項目がインプットデータになれば要件定義は終了する。

【0031】

このようにデータ項目の規定によって完了した要件定義は、そのデータ項目の並び方がデータの処理順序と全く関係が無い。従来、データの処理順序は、デー

タ項目を並べている順序或いは条件分岐や反復の指示等で人がプログラムによって指示してやらなければならなかった。本願第1発明によって行われるデータ項目による連鎖的機械的要件定義も、従来法のプログラムではその後人が処理順序を分析探索してそれをプログラムによって指示しなければならない。

【0032】

そこで本願に係る第2の発明（以下、「本願第2発明」ということもある。）は、人がプログラムによって処理順序を指示するのではなく、本願第1発明によって機械的に行われた要件定義をもとに、データ処理順序を自動的に決定する「プログラムの自動的開発手法」（例えば上記の「L y e e」で規定される手法）とを組み合わせることによって、結局最終製品たるソフトウェアの開発自体を自動的に行うものである。

【0033】

つまり本願第2発明によれば、本願第1発明を基礎とした上で、機械的に得られた要件定義をもとに、データの処理順序を自動的に見出すか、或いはデータを自動的に正しい順序で成立させた行くプログラム自動的開発手法、例えばL y e e、を用いるとするので、所望のソフトウェア開発が終局に至るまで自動化されるという劇的な効果もたらされるものである。本願第1発明については上記に詳述し、またL y e e発明については上記した文献に詳細に開示したので、ここではそれを指摘するにとどめ同じ説明の繰り返しは省略する。

【0034】

以上述べたことを要すれば、本願第1発明によれば、ソフトウェアを生産するにおいて、まず目的のデータ項目を決め、ついでその定義式や成立条件の中に新たに出て来た新しいデータ項目を更に規定することにより、人が処理順序等を考察することなく、データ項目の摘出及び要件定義を連鎖的機械的に簡単に行うことが可能となる。

【0035】

また、本願第2発明によれば、本願第1発明によって行われた機械的要件定義と、それらの規定されたデータ項目に基づいてデータの処理順序を自動的に見出して、或いはデータを自動的に正しい順序で成立させて行って、プログラムを自

動的に開発する手法（例えば上記の「L y e e」で規定される手法）とを用いるので、ソフトウェアの開発を自動的に行うことが可能となる。

【0036】

本発明の多くの特徴および利点は明細書の詳細な説明から明白である。更に、当該技術分野における通常の知識を有する者にとって修正および改変が容易に数多くなし得るので、図示および記述されたものと寸分違わぬ構成および動作に本発明を限定することは望ましくないことであり、従って、あらゆる適切な改変体および等価体は本発明の範囲に含まれるものと見なされうる。前述の本発明に係る実施形態の説明および例示によって詳細に記述されたが、添付の特許請求の範囲のみならず本発明に係る開示事項全体に定義された本発明の範囲から逸脱することなしに、修正、置換、および、変更が数多く可能である。

【0037】

また、本願に係る発明は、その適用において、上記の記述において説明されるか、或いは、図面に示された要素の詳細な解釈及び組み合わせに限定されるものではない。本発明は、他の実施形態が可能であり、種々の方法で実用および実施可能である。また、ここで用いられた語法および用語は記述を目的とするものであり、限定的に働くものとみなされてはならない。

【0038】

従って、当該技術分野における通常の知識を有する者は、本開示の基調となる概念は、本発明の幾つかの目的を実施するための他の構造、方法、及び、システムを設計するための基礎として容易に利用され得ることを理解するはずである。従って、本発明の趣旨および範囲から逸脱しない限り、本願の特許請求の範囲にはそのような等価な解釈が含まれるものと見なされるものである。

【0039】

また、上記ではコンピュータソフトウェアの自動開発法について主として説明したが、本発明に係る技術思想は、例えばコンピュータソフトウェアの自動開発装置、自動開発プログラム、自動開発プログラムを記録した記録媒体、自動開発プログラムを登載したコンピュータ・装置、自動開発プログラムを実行するクライアント・サーバ形式等といったカテゴリーにおいても実現、利用可能であるこ

とはいうまでもない。

【0040】

さらに本発明は、単一プロセッサ、単一ハードディスクドライブ、及び、単一ローカルメモリを備えたコンピュータシステムに限らず、当該システムのオプションとして、任意の複数または組み合わせプロセッサ又は記憶デバイスを装備するにも適している。コンピュータシステムは、精巧な計算器、掌タイプコンピュータ、ラップトップ／ノートブックコンピュータ、ミニコンピュータ、メインフレームコンピュータ、及び、スーパーコンピュータ、ならびに、これらの処理システムネットワーク組合わせを含む。本発明の原理に従って作動する任意の適切な処理システムによって代替されうるし、また、これらと組合せて用いることも可能である。

【0041】

また、本発明に係る技術思想は、もとよりあらゆる種類のプログラミング言語に対応可能である。さらに本発明に係る技術思想は、あらゆる種類・機能のアプリケーションソフトウェアに対しても適用可能である。

【0042】

また、本願第2発明については、本願第1発明と組み合わせる対象は、本願第1発明によって得られた機械的要件定義によってプログラムを自動的に開発する手法であればよく、上述の「L y e e」にその対象が限定されるわけではない。

【0043】

またさらに本願発明は、その技術思想の同一及び等価に及ぶ範囲において様々な変形、追加、置換、拡大、縮小等を許容するものである。また、本願発明を用いて生産されるソフトウェアが、その2次的生産品に登載されて商品化された場合であっても、本願発明の価値は何ら減ずるものではない。

【0044】

【発明の効果】

上述したように本発明の要件定義方法は、データ項目の摘出及び要件定義を連鎖的機械的に簡単に行うことが可能とする。

【0045】

また、本発明のコンピュータソフトウェアの自動開発方法は、コンピュータソフトウェアの開発工程から人間が必ず指示しなければならないもの以外をなくし、最終的にソフトウェアの開発を極力機械的にし、自動化を可能にする。

【図面の簡単な説明】

【図 1】

データ項目間の関係を模式的に示すと共に、連鎖式データ項目規定による要件定義法 (Data Chain Requirement Definition (DCRD) 法) を概念的に説明するための概念図である。

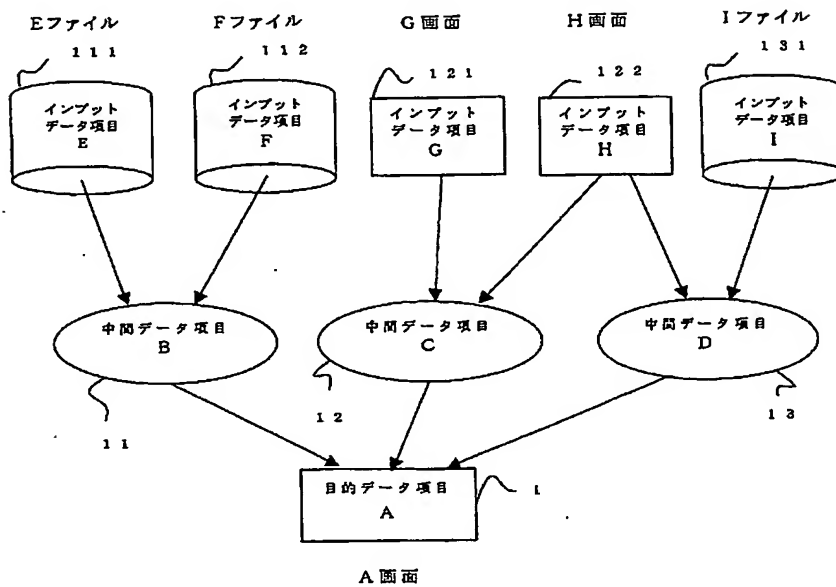
【符号の説明】

- 1 目的データ項目 A 及び画面 A
- 1 1 中間データ項目 B
- 1 2 中間データ項目 C
- 1 3 中間データ項目 D
- 1 1 1 インプットデータ項目 E 及び E ファイル
- 1 1 2 インプットデータ項目 F 及び F ファイル
- 1 2 1 インプットデータ項目 G 及び G 画面
- 1 2 2 インプットデータ項目 H 及び H 画面
- 1 3 1 インプットデータ項目 I 及び I ファイル

【書類名】 図面

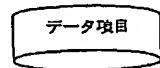
【図 1】

(この図はデータ処理の順序を示すデータフローチャートとは全く異なる。この図はデータ項目間の関係を示すだけであることを注意。)

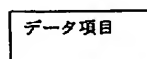


凡例

上図のデータ項目 A, B, C, D の関係は、A というデータ項目は B データ項目, C データ項目, D データ項目から成ることを示す。



ファイルの中にデータ項目がある。



画面の中にデータ項目がある。

【書類名】 要約書

【要約】

【課題】

機械的に要件定義を完了するのを可能にすること、さらに、ソフトウェアの開発を自動的、且つ、従来法に比較して革命的に高効率で行うことを可能にすること。

【解決手段】 開発対象のコンピュータソフトウェアが最終的に求めようとするアウトプット項目 1 を取り出し、これらを中間データ項目 1 1、1 2 及び 1 3 を用いた計算式により規定し、これらの中間データ項目の総てを、それぞれ別の計算式により規定し、このようなことを、当該計算式を構成するのがインプットデータ項目のみとなるまで繰り返し、この結果得られたデータ項目及びその成立条件等の規定を要件定義とする。さらにこうして得られた要件定義と、それに基づいてデータの処理順序を自動的に見出して、或いはデータを自動的に正しい順序で成立させて行って、プログラムを自動的に開発する方法を適用することによって、最終的に求めるソフトウェアの自動的开发を行う。

【選択図】 図 1

特願 2 0 0 2 - 2 6 2 6 7 0

出 願 人 履 歴 情 報

識別番号

[5 9 9 0 8 6 2 3 8]

1. 変更年月日
[変更理由]

住 所
氏 名

1 9 9 9 年 6 月 2 1 日
新規登録
東京都港区高輪 3 丁目 1 1 番 3 号
ソフトウェア生産技術研究所株式会社

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☒ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☒ **FADED TEXT OR DRAWING**
- ☒ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.